```
#profile and concensus
sequences = ['ATATAAAAAA',
          'ATATAAACCC',
          'ATATAAATCG',
          'ATATATGTGT',
          'ATATAAGTCC']

profile = []
ac, cc, gc, tc = [], [], [], []

#length of sequences
n = len(sequences[0])
#number of sequences
m = len(sequences)

concensus = ''
for i in range(n):
   alignment = ''
   for j in range(m):
      alignment += sequences[j][i]
   print alignment
   a = alignment.count('A')
   c = alignment.count('C')
   g = alignment.count('G')
   t = alignment.count('T')
   mc = max(a, c, g, t)
   #print mc
   if(mc == a): concensus += 'A'
   elif (mc == c): concensus += 'C'
   elif (mc == g): concensus += 'G'
   else: concensus += 'T'

   ac.append(a)
   cc.append(c)
   gc.append(g)
   tc.append(t)

print concensus
print 'A:', ac
print 'C:', cc
print 'G:', gc
print 'T:', tc
```

```
#dotplot
dotplot = []

s1 = 'ATATACG'
s2 = 'ATACGCGATA'

dotplot.append(' ' +s2)
for i in s1:
   row = []
   row.append(i)
   for j in s2:
      if i == j:
         row.append('*')
      else:
         row.append(' ')
   dotplot.append(''.join(row))

for row in dotplot:
   print row
```

```
#k-mer
seq = 'ACATTTATCA'
k = 3
n = len(seq)

for i in range(n - k + 1):
   print seq[i:i+k]
```

```
#time
import time

start = time.time()
for i in range(100000000):
   pass
end = time.time()
print 'time ', (end-start)
```

```
#longest orf
s1 =
'AAAAUGGAUGGACAAUAUAAUGAUUUUUAA'

start = 'AUG'
stop = ['UAG', 'UAA', 'UGA']

sp, se = -1, -1
maxl, maxr = 0, ''
n = len(s1)
for j in range(3):
    for i in range(j, n-2, 3):
        codon = s1[i: i+3]    #print codon
        if codon == start:
            sp = i    #print 'codon: ', codon
        if codon in stop:
            #print 'codon: ', codon
            se = i+3
            if sp >=0:
                length = se - sp
                print s1[sp:se], length
                if length > maxl:
                    maxl = length
                    maxr = s1[sp:se]
                    sp, se = -1, -1
    sp, se = -1, -1
print '', maxr, maxl
```

```
#protein digest
seq =
'MRHIAHTQRCLSRLTSLVALLLIVLPMVFSPAHSCGPGR
GLGRHRARNLYPLVLKQTIPNLSEYTNSASGPLEGVIRRD
SPKFKDLVPNYNRDILFRDEE'
c = 'L'
peptides = seq.split(c)
peptides = [p for p in peptides if p]
print peptides


#########################
#alternative way
peptides = []
peptide = ''
for p in seq:
    if p == c:
        if peptide: peptides.append(peptide)
        peptide = ''
    else:
        peptide += p

if peptide : peptides.append(peptide)
print peptides
```

```
#read/wrire csv
import csv

data = []
with open('SampleCSVFile.csv', 'r') as csvf:
    csvreader = csv.reader(csvf)
    for row in csvreader:
        #print len(row)
        data.append(row)

print data
with open('MyCSVOutput.csv', 'w') as csvf:
    csvwriter = csv.writer(csvf)
    for row in data:
        csvwriter.writerow(row[:5])
```

```
#read xml
from xml.etree import ElementTree as et

with open('SampleXMLFile.xml', 'r') as xmlf:
    tree = et.parse(xmlf)

    for entry in tree.iter():
        print entry.tag, entry.attrib, entry.text

'''
For detailed example visit
https://docs.python.org/3/library/xml.etree.eleme
nttree.html
'''
```